WHAT IS CLAIMED IS:

1   1.   A method for interrupt processing, comprising:

2   determining that an event has occurred;

3   determining a processor identifier;

4   determining an event data structure identifier for an event data structure into

5   which data for the event is stored using the processor identifier;

6   determining a vector identifier for an interrupt message vector; and

7   writing interrupt message data to the interrupt message vector to generate an

8   interrupt.

1   2.   The method of claim 1, wherein the processor identifier is determined by

2   applying a hash technique to a data packet to access a processor redirection/indirection

3   structure.

1   3.   The method of claim 1, wherein the event data structure identifier is

2   determined by accessing a message vector mapping structure using the processor

3   identifier and an event code.

1   4.   The method of claim 1, wherein the vector identifier is determined by

2   accessing a message vector mapping structure using the processor identifier and an event

3   code.

1   5.   The method of claim 1, wherein the event data structure identifier is

2   determined by accessing a processor redirection/indirection structure using the processor

3   identifier.

1   6.   The method of claim 5, wherein the vector identifier is determined from a

2   message vector mapping structure using the event data structure identifier as an index.

1   7.   The method of claim 1, further comprising:

2   writing an event entry to the event data structure identified by the event data

3   structure identifier; and

4   advancing a write indicator.

1   8.   The method of claim 1, further comprising:

2   receiving an interrupt;

3   identifying an event data structure using the interrupt message data in the interrupt

4   message vector; and

5   processing an event entry in the identified event data structure.

1   9.   The method of claim 1, further comprising:

2   determining whether the event is associated with data; and

3   determining a default processor identifier in response to determining that the

4   event is not associated with data.

1   10.   A system for in interrupt processing, comprising:

2   an Input/Output device coupled to a bus; and

3   circuitry at the Input/Output device operable to:

4   determine that an event has occurred;

5   determine a processor identifier from a processor redirection/indirection

6   structure;

7   determine an event data structure identifier for an event data structure into

8   which data for the event is stored using the processor identifier;

9   determine a vector identifier for an interrupt message vector into which an

10   interrupt message is written; and

11   write interrupt message data to the interrupt message vector to generate an

12   interrupt.

1   11.   The system of claim 11, wherein the processor identifier is determined by

2   applying a hash technique to a data packet to access a processor redirection/indirection

3   structure.

1   12.   The system of claim 11, wherein the event data structure identifier is
2   determined by accessing a message vector mapping structure using the processor
3   identifier and an event code.

1   13.   The system of claim 11, wherein the vector identifier is determined by
2   accessing a message vector mapping structure using the processor identifier and an event
3   code.

1   14.   The system of claim 11, wherein the event data structure identifier is
2   determined by accessing a processor redirection/indirection structure using the processor
3   identifier.

1   15.   The system of claim 15, wherein the vector identifier is determined from a
2   message vector mapping structure using the event data structure identifier as an index.

1   16.   The system of claim 11, wherein the circuitry is operable to:
2   write an event entry to the event data structure identified by the event data
3   structure identifier; and
4   advance a write indicator.

1   17.   The system of claim 11, further comprising:
2   an Input/Output device driver coupled to a bus; and
3   circuitry at the Input/Output device driver operable to:
4   receive an interrupt;
5   identify an event data structure using the interrupt message data in the
6   interrupt message vector; and
7   process an event entry in the identified event data structure.

1   18.   The system of claim 11, wherein the circuitry is operable to:
2   determine whether the event is associated with data; and

3   determine a default processor identifier in response to determining that the event

4 is not associated with data.

1   19.   An article of manufacture for interrupt processing, wherein the article of

2 manufacture is at an Input/Output device and is operable to:

3   determine that an event has occurred;

4   determine a processor identifier from a processor redirection/indirection structure;

5   determine an event data structure identifier for an event data structure into which

6 data for the event is stored using the processor identifier;

7   determine a vector identifier for an interrupt message vector into which an

8 interrupt message for the event is stored; and

9   write interrupt message data to the interrupt message vector to generate an

10 interrupt

1   20.   The article of manufacture of claim 19, wherein the processor identifier is

2 determined by applying a hash technique to a data packet to access a processor

3 redirection/indirection structure.

1   21.   The article of manufacture of claim 19, wherein the event data structure

2 identifier is determined by accessing a message vector mapping structure using the

3 processor identifier and an event code.

1   22.   The article of manufacture of claim 19, wherein the vector identifier is

2 determined by accessing a message vector mapping structure using the processor

3 identifier and an event code.

1   23.   The article of manufacture of claim 19, wherein the event data structure

2 identifier is determined by accessing processor redirection/indirection structure using the

3 processor identifier.

1   24.   The article of manufacture of claim 23, wherein the vector identifier is

2   determined from a message vector mapping structure using the event data structure

3   identifier as an index.

1   25.   The article of manufacture of claim 19, wherein the article of manufacture

2   is operable to:

3   write an event entry to the event data structure identified by the event data

4   structure identifier; and

5   advance a write indicator.

1   26.   The article of manufacture of claim 19, wherein the Input/Output device is

2   connected to a device driver and wherein an article of manufacture at the Input/Output

3   device driver is operable to:

4   receive an interrupt;

5   identify an event data structure using the interrupt message data in the interrupt

6   message vector; and

7   process an event entry in the identified event data structure.

1   27.   The article of manufacture of claim 26, wherein the article of manufacture

2   is operable to:

3   determine whether the event is associated with data; and

4   determine a default processor identifier in response to determining that the event

5   is not associated with data.